



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського”

Факультет прикладної математики
Кафедра програмного забезпечення комп’ютерних систем

Лабораторна робота № 2
з дисципліни “ Математичні та алгоритмічні основи комп’ютерної
графіки”

Виконав
студент III курсу
групи КП-73

Булаєвський Ігор Олегович
(прізвище, ім'я, по батькові)

Варіант №3

Київ - 2020

Лістинг програми

Animation.java

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.geom.GeneralPath;
import javax.swing.JFrame;
import javax.swing.Timer;

public class Animation extends Skeleton implements ActionListener {
    double[][] points = {
        { -100, -15 }, { -25, -25 }, { 0, -90 }, { 25, -25 },
        { 100, -15 }, { 50, 25 }, { 60, 100 }, { 0, 50 },
        { -60, 100 }, { -50, 25 }, { -100, -15 }
    };
    Timer timer;
    private GeneralPath star;

    private double angle = 0;
    private double dx = -150;
    private double dy = 0;

    public Animation() {
        star = pathFromPoints(points);
        timer = new Timer(10, this);
        timer.start();
    }

    public void paint(Graphics g) {
        super.paint(g);
        g2d.translate(width / 2, 0);
        drawFrame();
        // Центр координат в середину рамки
        g2d.translate(width / 4, height / 2);
        // Для обертання навколо кута
        g2d.rotate(angle, points[0][0], points[0][1]);
        g2d.setColor(Color.GREEN);
        g2d.fill(star);
        g2d.rotate(-angle, points[0][0], points[0][1]);

        // Для руху по колу
        g2d.translate(dx, dy);
        g2d.setColor(Color.PINK);
        g2d.fill(star);
    }

    public void drawFrame() {
```

```

        int frameWeight = 16;
        BasicStroke bs = new BasicStroke(frameWeight, BasicStroke.CAP_ROUND,
            BasicStroke.JOIN_ROUND);
        g2d.setStroke(bs);
        g2d.setColor(Color.RED);
        g2d.drawRect(frameWeight / 2, frameWeight / 2,
            width / 2 - frameWeight, height - frameWeight);
    }

    public void actionPerformed(ActionEvent e) {
        angle += 0.07;
        double x = -150;
        double y = 0;
        double cos = Math.cos(-angle);
        double sin = Math.sin(-angle);
        dx = x * cos - y * sin;
        dy = x * sin + y * cos;
        repaint();
    }

    public static void main(String[] args) {
        JFrame frame = new JFrame("CGI LAB2");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        width = 1600;
        frame.setSize(width, height);
        frame.setLocationRelativeTo(null);
        frame.setResizable(false);
        frame.add(new Animation());
        frame.setVisible(true);

        Dimension size = frame.getSize();
        Insets insets = frame.getInsets();
        height = size.height - insets.top - insets.bottom;
    }
}

```

Skeleton.java

```

import java.awt.geom.AffineTransform;
import java.awt.geom.GeneralPath;
import java.awt.geom.Rectangle2D;
import java.awt.image.ColorModel;
import javax.swing.JFrame;
import javax.swing.JPanel;

@SuppressWarnings("serial")
public class Skeleton extends JPanel {

```

```

protected static int width = 800;
protected static int height = 500;
protected Graphics2D g2d;

public void paint(Graphics g) {
    g2d = (Graphics2D) g;
    setRenderHints();
    setBackgroundColor(new Color(100, 100, 100));
    drawPillar();
    drawSign();
    drawLights();
}

private void setRenderHints() {
    RenderingHints rh = new RenderingHints(
        RenderingHints.KEY_ANTIALIASING,
        RenderingHints.VALUE_ANTIALIAS_ON);
    rh.put(RenderingHints.KEY_RENDERING,
        RenderingHints.VALUE_RENDER_QUALITY);
    g2d.setRenderingHints(rh);
}

private void setBackgroundColor(Color color) {
    g2d.setBackground(color);
    g2d.clearRect(0, 0, width, height);
}

private void drawPillar() {
    GradientPaint gp = new GradientPaint(5, 10,
        Color.BLACK, 10, 5, Color.LIGHT_GRAY, true);
    g2d.setPaint(gp);
    g2d.fillRect(385, 260, 30, 200);
}

protected GeneralPath pathFromPoints(double[][] points) {
    GeneralPath path = new GeneralPath();
    path.moveTo(points[0][0], points[0][1]);
    for (int k = 1; k < points.length; k++)
        path.lineTo(points[k][0], points[k][1]);
    path.closePath();
    return path;
}

private void drawSign() {
    // Red part
    double[][] pointsRedBoard = {{ 285, 260 }, { 515, 260}, { 400, 40 }};

```

```

        GeneralPath redBoard = pathFromPoints(pointsRedBoard);
        g2d.setColor(Color.RED);
        g2d.fill(redBoard);

        // White part
        double[][] pointsWhiteBoard = {{ 315.0, 240.0 }, { 485.0, 240.0 }, { 400.0, 75.0
    }};

        GeneralPath whiteBoard = pathFromPoints(pointsWhiteBoard);
        g2d.setColor(Color.WHITE);
        g2d.fill(whiteBoard);
    }

    private void drawLights() {
        g2d.setColor(Color.RED);
        g2d.fillOval(380,105,40, 40);
        g2d.setColor(Color.YELLOW);
        g2d.fillOval(380,150,40, 40);
        g2d.setColor(Color.GREEN);
        g2d.fillOval(380,195,40, 40);
    }

    public static void main(String[] args) {
        JFrame frame = new JFrame("CGI LAB2");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(width, height);
        frame.setLocationRelativeTo(null);
        frame.setResizable(false);
        frame.add(new Skeleton());
        frame.setVisible(true);
    }
}

```

Скріншоти результатів

